BA

(12) UK Patent Application (19) GB (11) 2 332 544 (13) A

(43) Date of A Publication 23.06.1999

(21) Application No 9827135.6

(22) Date of Filing 09.12.1998

(30) Priority Data
(31) 08995616 (32) 22.12.1997 (33) US

(51) INT CL[6]
G06F 17/30

(52) UK CL (Edition Q)
G4A AUXX

(56) Documents Cited
EP 0802492 A1 EP 0762297 A2 EP 0378848 A2
US 5404295 A

(58) Field of Search
UK CL (Edition Q) G4A AUDB AUXX
INT CL[6] G06F

(71) Applicant(s)
Ricoh Company Limited
(Incorporated in Japan)
No 3-6 Nakamagome 1-chome, Ohta-ku, Tokyo, Japan

(72) Inventor(s)
Jamey Graham
David G. Stork

(74) Agent and/or Address for Service
Marks & Clerk
57-60 Lincoln's Inn Fields, LONDON, WC2A 3LS,
United Kingdom

(54) Abstract Title
Automatic adaptive document help system

(57) In an automatic reading assistance system for electronic documents (502), an automatic annotator (508) finds concepts of interest (512) and keywords. The operation of the annotator is personalizable (518) for a particular user. The annotator is also capable of improving its performance over time by both automatic and manual feedback. Another available feature is a thumbnail image of all or part of a multi-page document wherein a currently displayed section of the document is highlighted in the thumbnail image. Movement of the highlighted area in the thumbnail image is then synchronized with scrolling through the document.

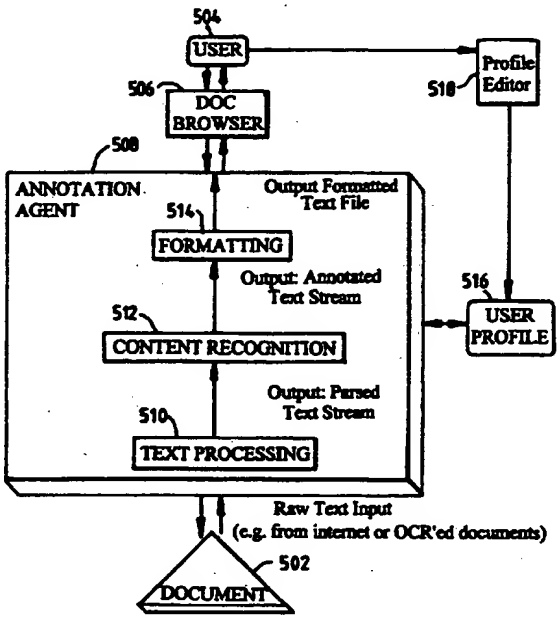


Fig. 5

GB 2 332 544 A

At least one drawing originally filed was informal and the print reproduced here is taken from a later filed formal copy.

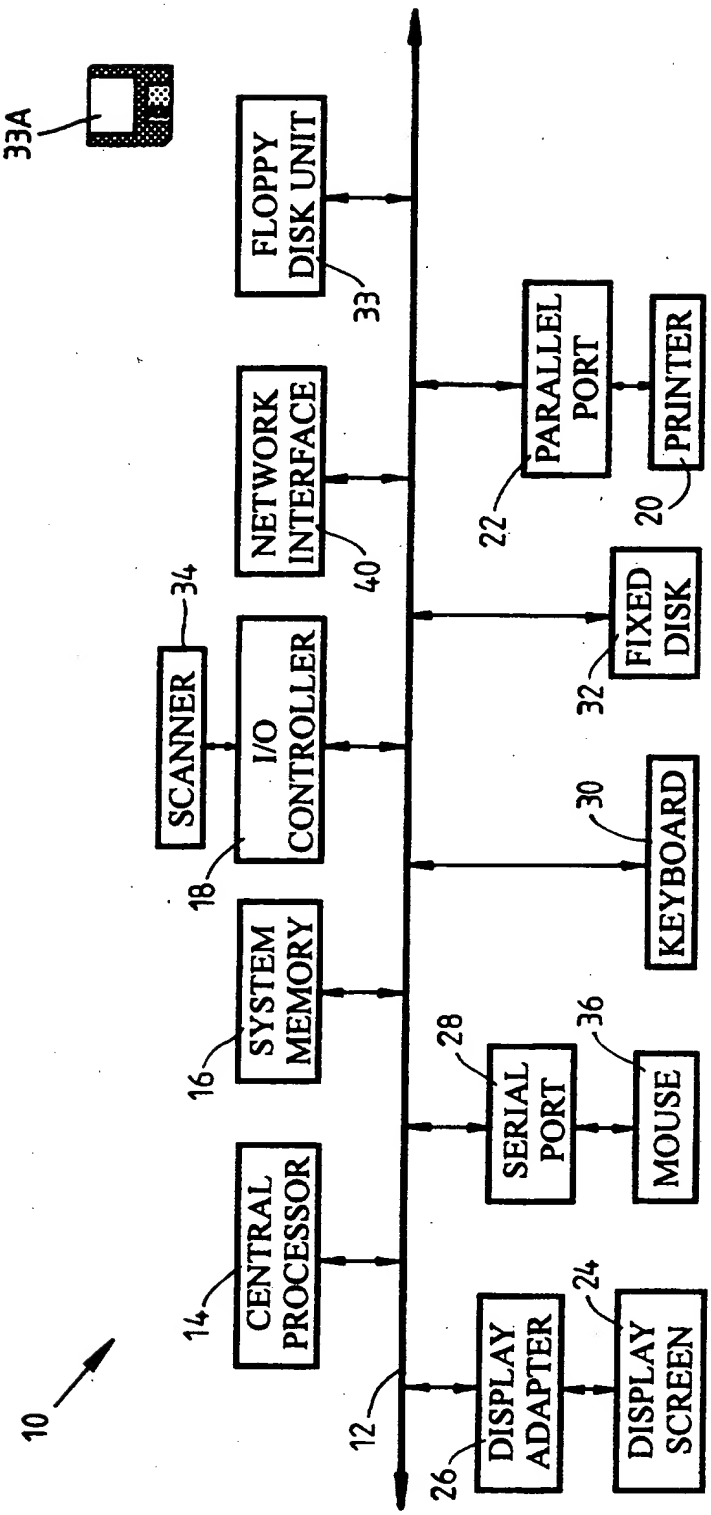

10
12
14
CENTRAL PROCESSOR
16
SYSTEM MEMORY
18
I/O CONTROLLER
20
PRINTER
22
PARALLEL PORT
24
DISPLAY SCREEN
26
DISPLAY ADAPTER
28
SERIAL PORT
30
KEYBOARD
32
FIXED DISK
33
FLOPPY DISK UNIT
33A
34
SCANNER
36
MOUSE
40
NETWORK INTERFACE

Fig. 1

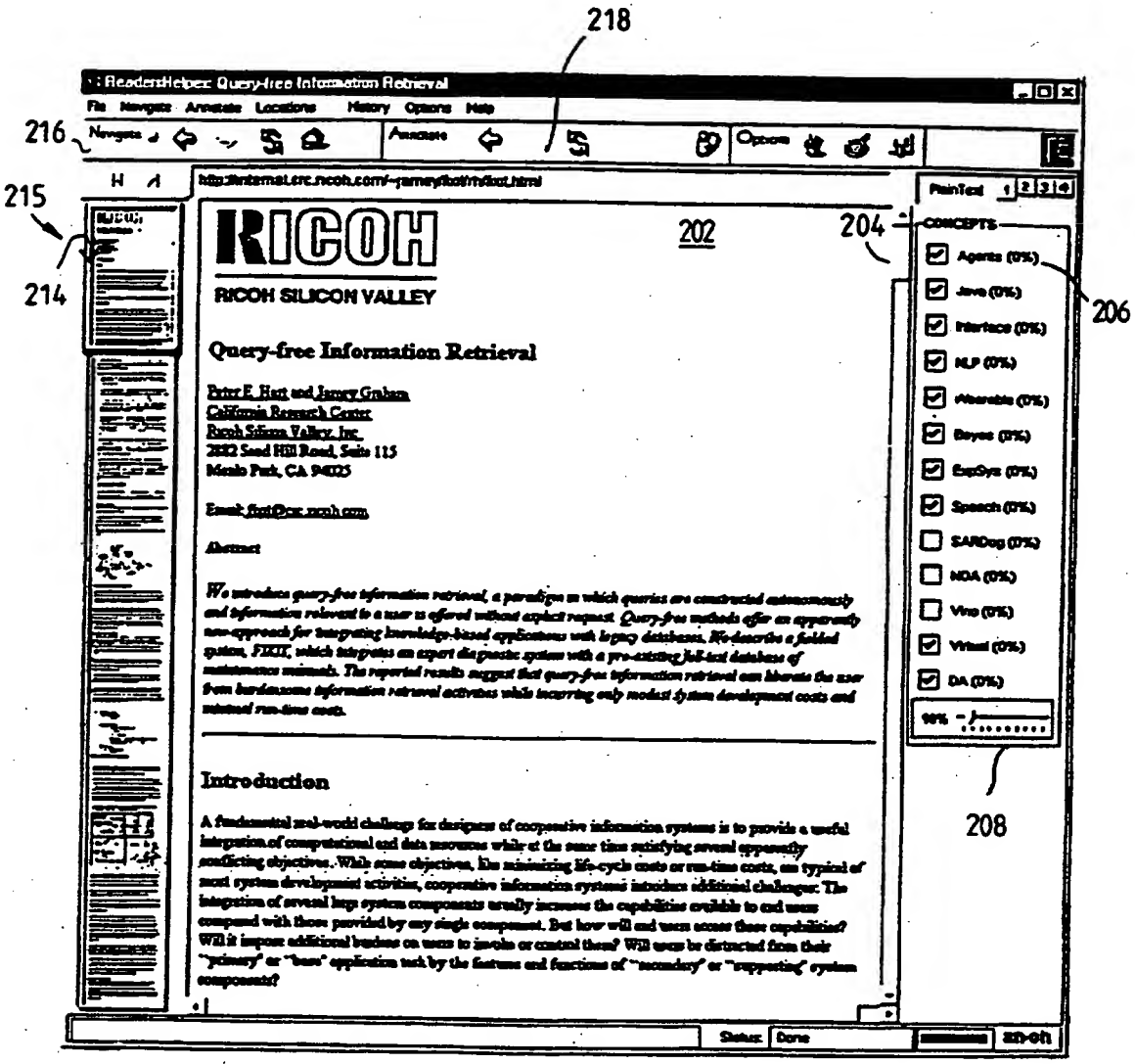

218
216
215
214
204
206
208
202
RICOH
RICOH SILICON VALLEY
Query-free Information Retrieval
Introduction
CONCEPTS

Fig. 2A

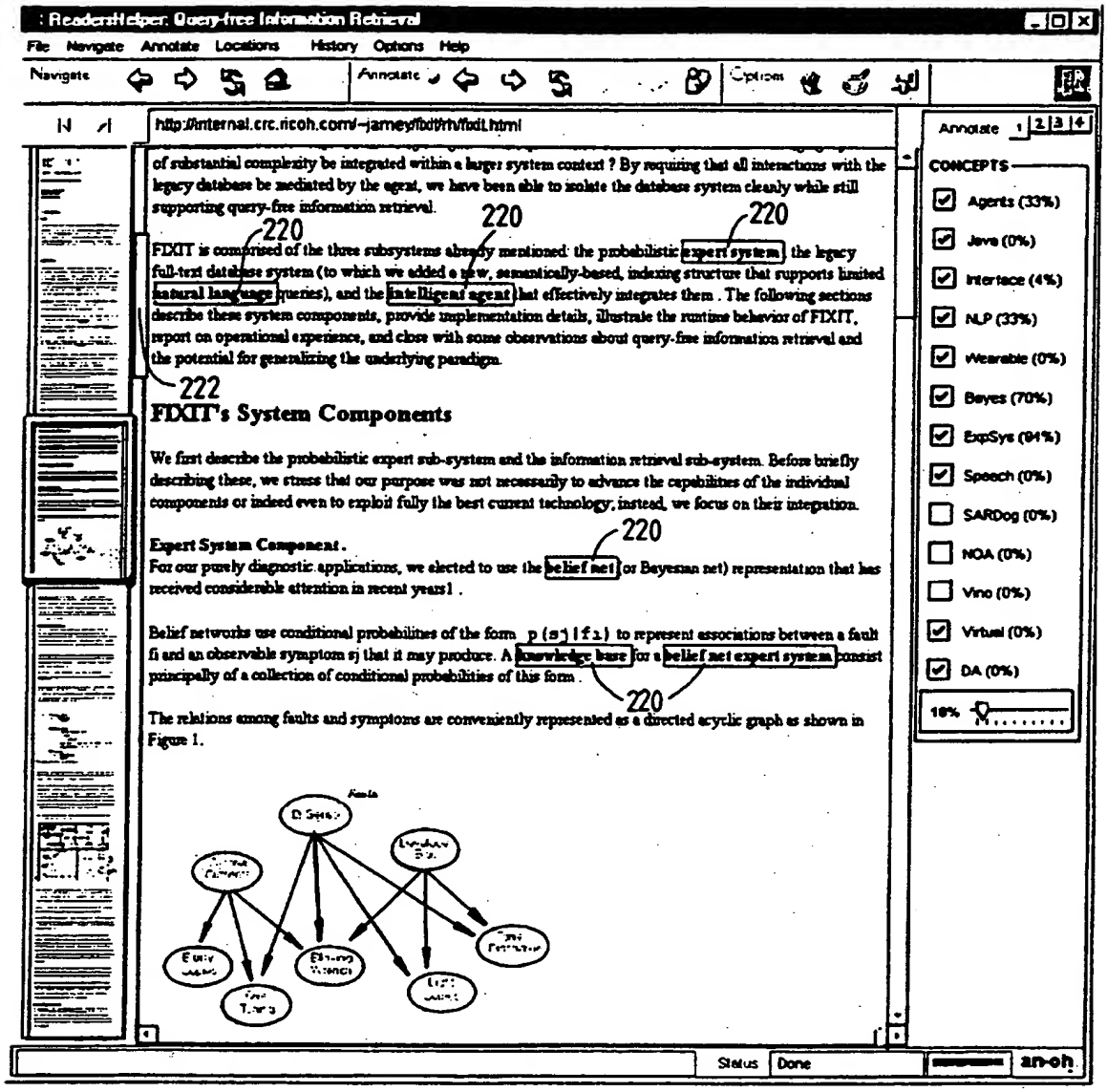

ReaderHelper: Query-free Information Retrieval
File Navigate Annotate Locations History Options Help
Navigate
Annotate
Options
http://internal.crc.ricoh.com/~jamey/fxit/html/fxit.html
of substantial complexity be integrated within a larger system context ? By requiring that all interactions with the legacy database be mediated by the agent, we have been able to isolate the database system cleanly while still supporting query-free information retrieval.
220
220
220
FIXIT is comprised of the three subsystems already mentioned: the probabilistic expert system, the legacy full-text database system (to which we added a new, semantically-based, indexing structure that supports limited natural language queries), and the intelligent agent that effectively integrates them . The following sections describe these system components, provide implementation details, illustrate the runtime behavior of FIXIT, report on operational experience, and close with some observations about query-free information retrieval and the potential for generalizing the underlying paradigm.
222
FIXIT's System Components
We first describe the probabilistic expert sub-system and the information retrieval sub-system. Before briefly describing these, we stress that our purpose was not necessarily to advance the capabilities of the individual components or indeed even to exploit fully the best current technology; instead, we focus on their integration.
Expert System Component.
220
For our purely diagnostic applications, we elected to use the belief net (or Bayesian net) representation that has received considerable attention in recent years1 .
Belief networks use conditional probabilities of the form p(sj|fi) to represent associations between a fault fi and an observable symptom sj that it may produce. A knowledge base for a belief net expert system consist principally of a collection of conditional probabilities of this form .
220
The relations among faults and symptoms are conveniently represented as a directed acyclic graph as shown in Figure 1.
Annotate 1 2 3 4
CONCEPTS
Agents (33%)
Java (0%)
Interface (4%)
NLP (33%)
Wearable (0%)
Bayes (70%)
ExpSys (94%)
Speech (0%)
SARDog (0%)
NOA (0%)
Vino (0%)
Virtual (0%)
DA (0%)
18%
Status Done

Fig. 2B

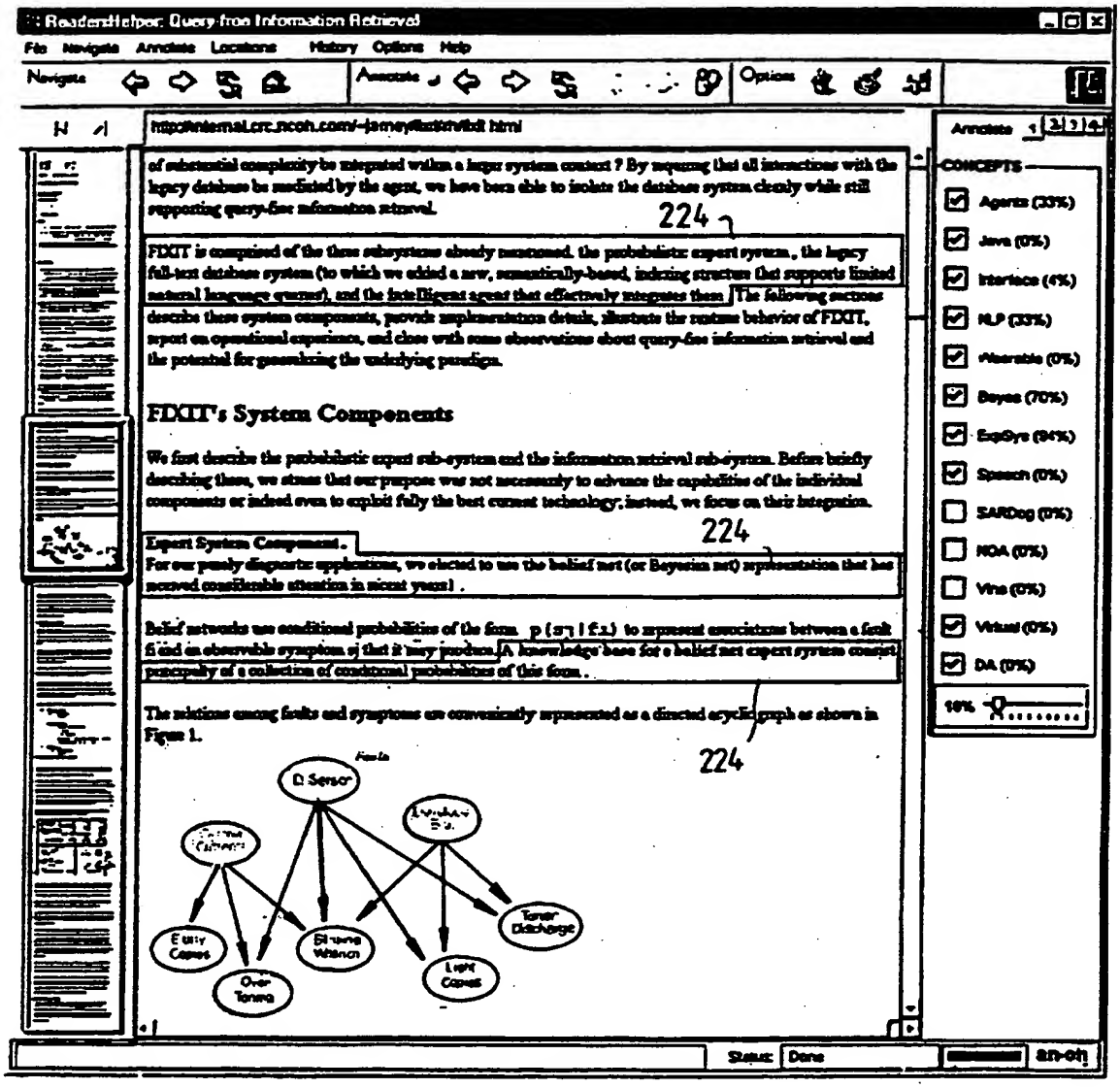

ReadersHelper: Query-free Information Retrieval
File Navigate Annotate Locations History Options Help
Navigate
Annotate
Options
224
FDIT's System Components
224
224
Expert System Component.
Status: Done

Fig. 2C

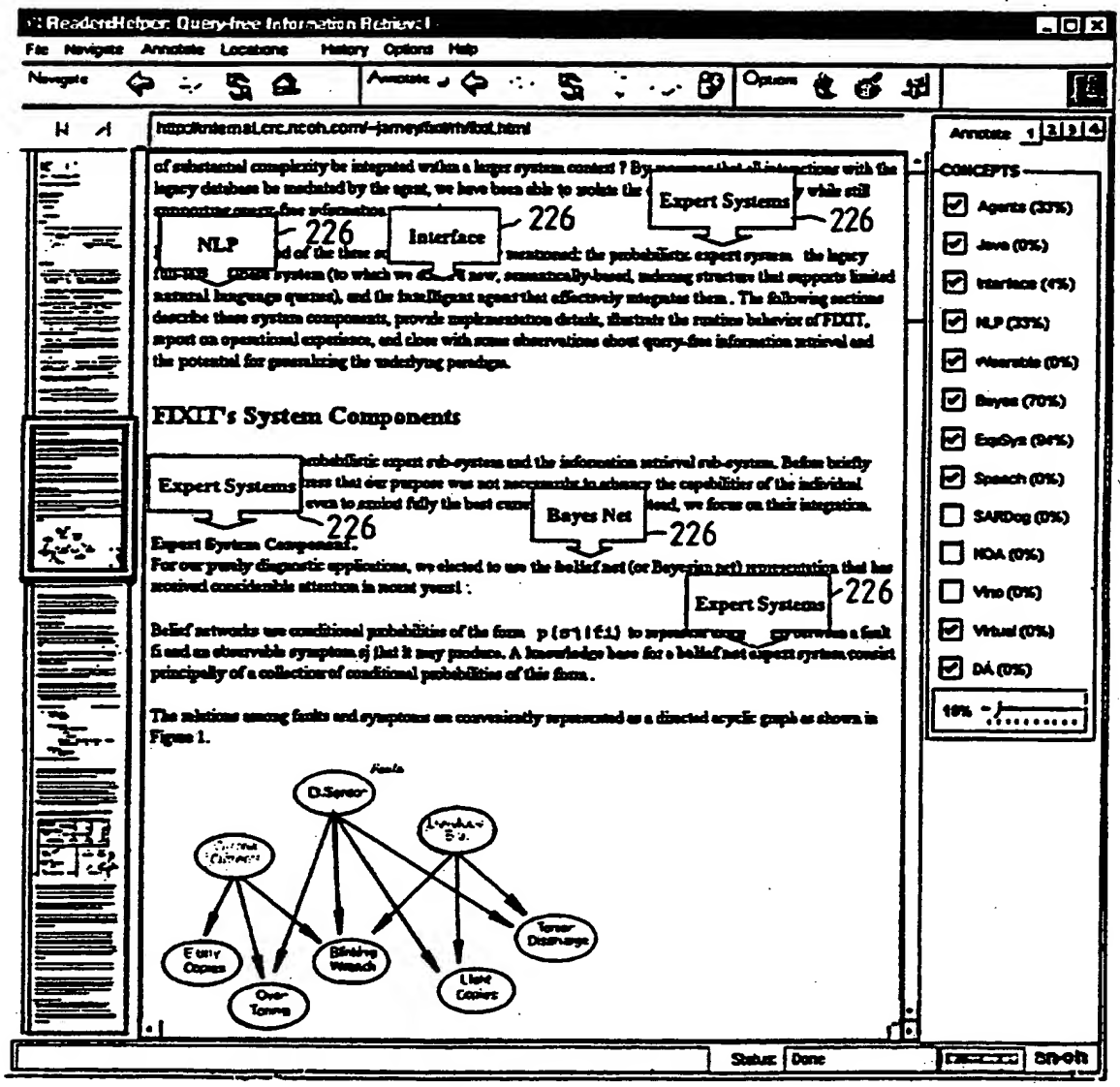

ReaderHelper: Query-free Information Retrieval
File Navigate Annotate Locations History Options Help
Navigate
Annotate
Options
Expert Systems
226
NLP
226
Interface
226
Expert Systems
226
FIXIT's System Components
Expert Systems
226
Bayes Net
226
Expert System Components:
Expert Systems
226
Annotate 1 2 3 4
CONCEPTS
Agents (33%)
Java (0%)
Interface (4%)
NLP (33%)
Heuristic (0%)
Bayes (70%)
ExpSys (94%)
Speech (0%)
SARDog (0%)
NOA (0%)
Vino (0%)
Virtual (0%)
DA (0%)
Status: Done

Fig. 2D

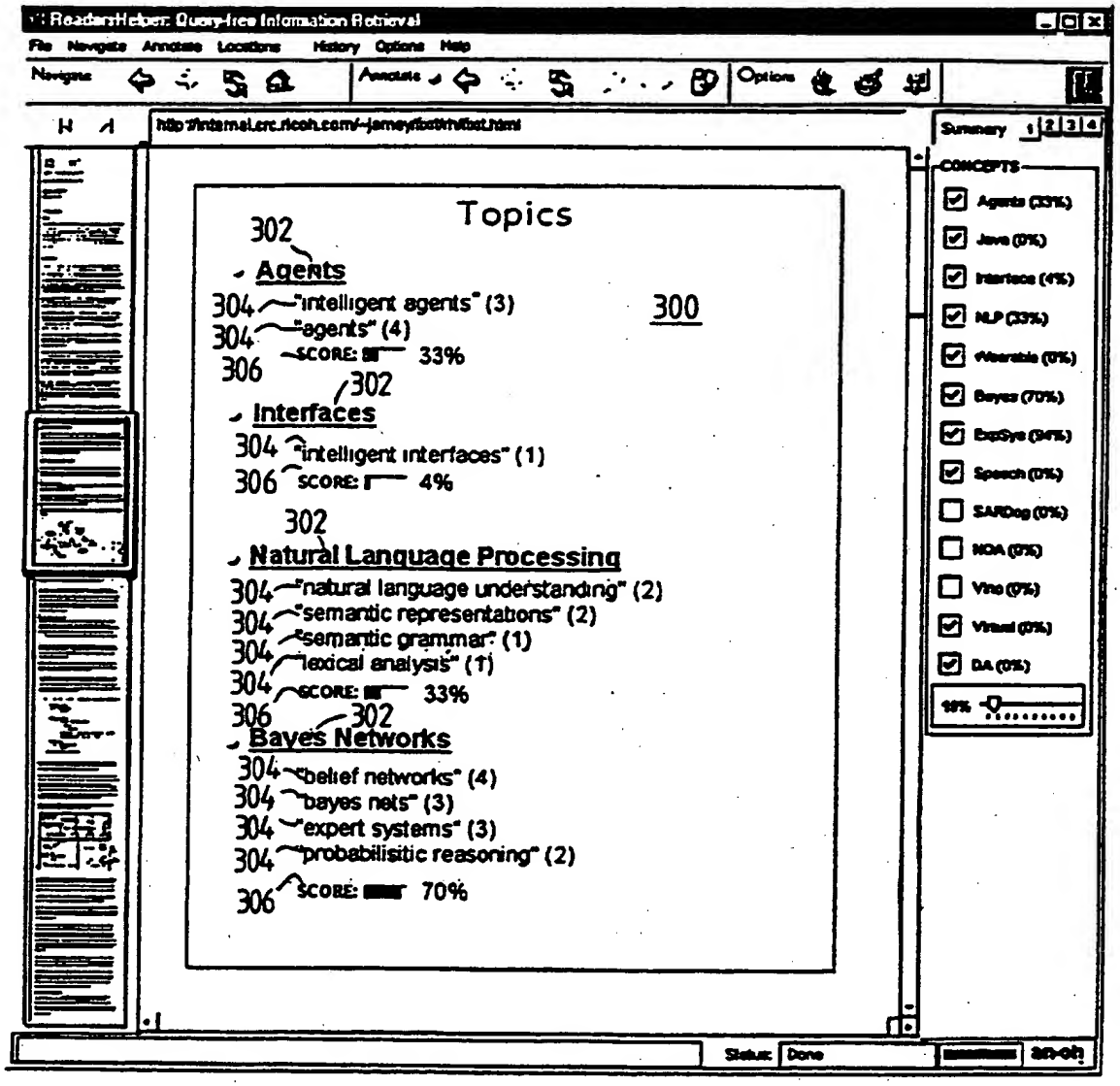

Topics
302
Agents
304 "intelligent agents" (3)
304 "agents" (4)
306 SCORE: 33%
300
302
Interfaces
304 "intelligent interfaces" (1)
306 SCORE: 4%
302
Natural Language Processing
304 "natural language understanding" (2)
304 "semantic representations" (2)
304 "semantic grammar" (1)
304 "lexical analysis" (1)
304 SCORE: 33%
306
302
Bayes Networks
304 "belief networks" (4)
304 "bayes nets" (3)
304 "expert systems" (3)
304 "probabilisitic reasoning" (2)
306 SCORE: 70%
Summary
CONCEPTS
Agents (33%)
Java (0%)
Interface (4%)
NLP (33%)
Wearable (0%)
Bayes (70%)
ExpSys (34%)
Speech (0%)
SARDog (0%)
NOA (0%)
Vino (0%)
Virtual (0%)
DA (0%)
Status: Done

Fig. 3

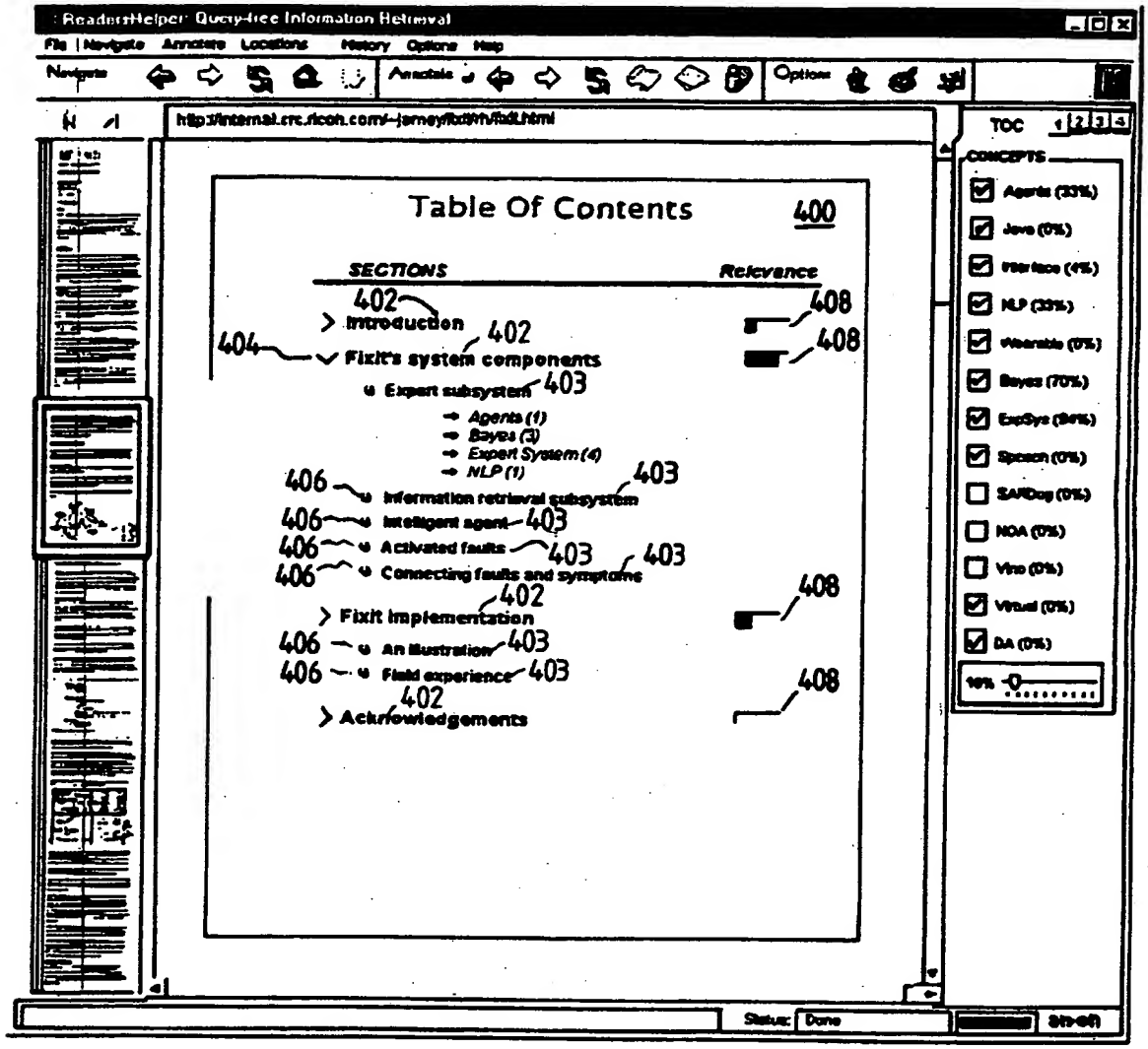

ReadersHelper: Query-free Information Retrieval
Table Of Contents
400
SECTIONS
Relevance
402
Introduction
408
404
Fixit's system components
402
408
Expert subsystem
403
Agents (1)
Bayes (3)
Expert System (4)
NLP (1)
406
Information retrieval subsystem
403
406
Intelligent agent
403
406
Activated faults
403
403
406
Connecting faults and symptoms
Fixit implementation
402
408
406
An illustration
403
406
Field experience
403
402
408
Acknowledgements
TOC
CONCEPTS
Status:
Done

Fig. 4

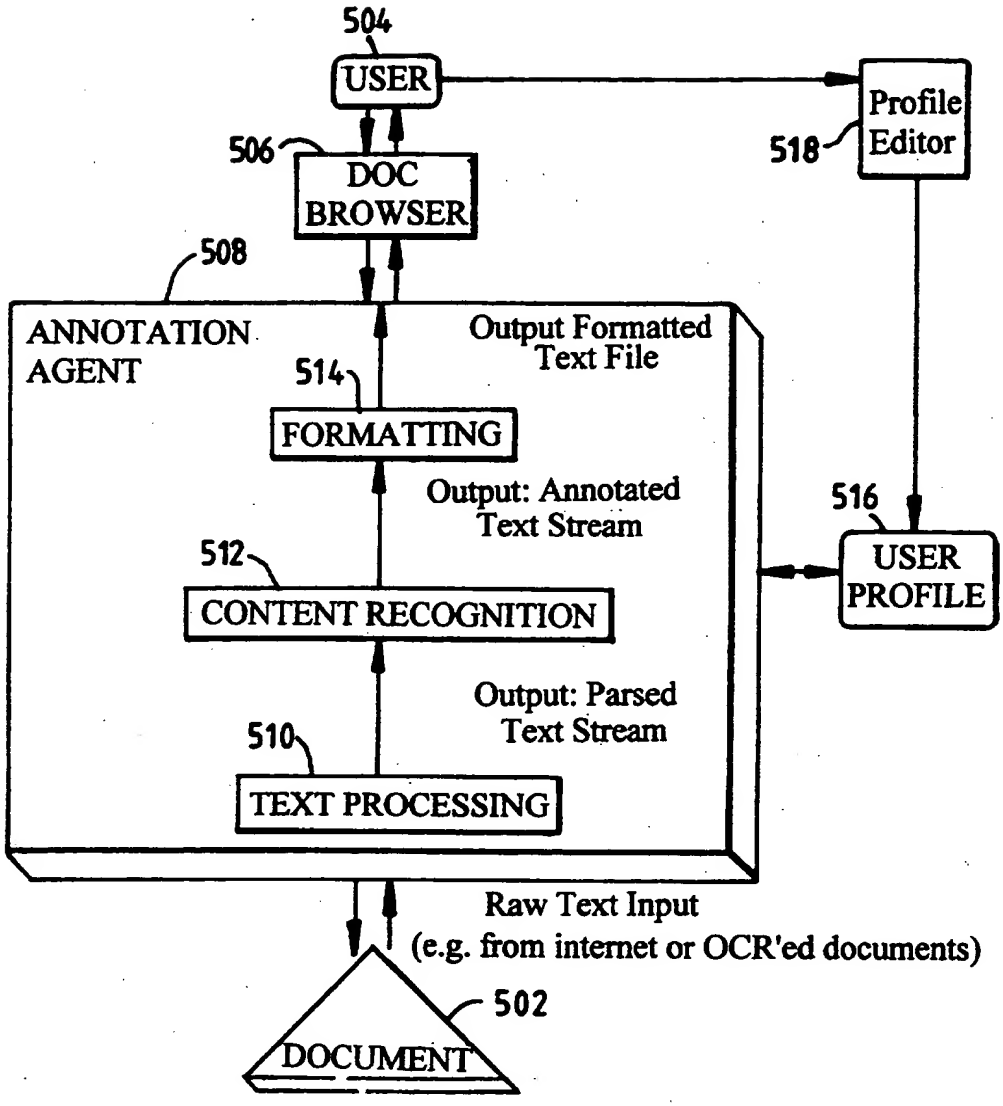

504
USER
Profile Editor
518
506
DOC BROWSER
508
ANNOTATION AGENT
Output Formatted Text File
514
FORMATTING
Output: Annotated Text Stream
516
USER PROFILE
512
CONTENT RECOGNITION
Output: Parsed Text Stream
510
TEXT PROCESSING
Raw Text Input
(e.g. from internet or OCR'ed documents)
502
DOCUMENT

Fig. 5

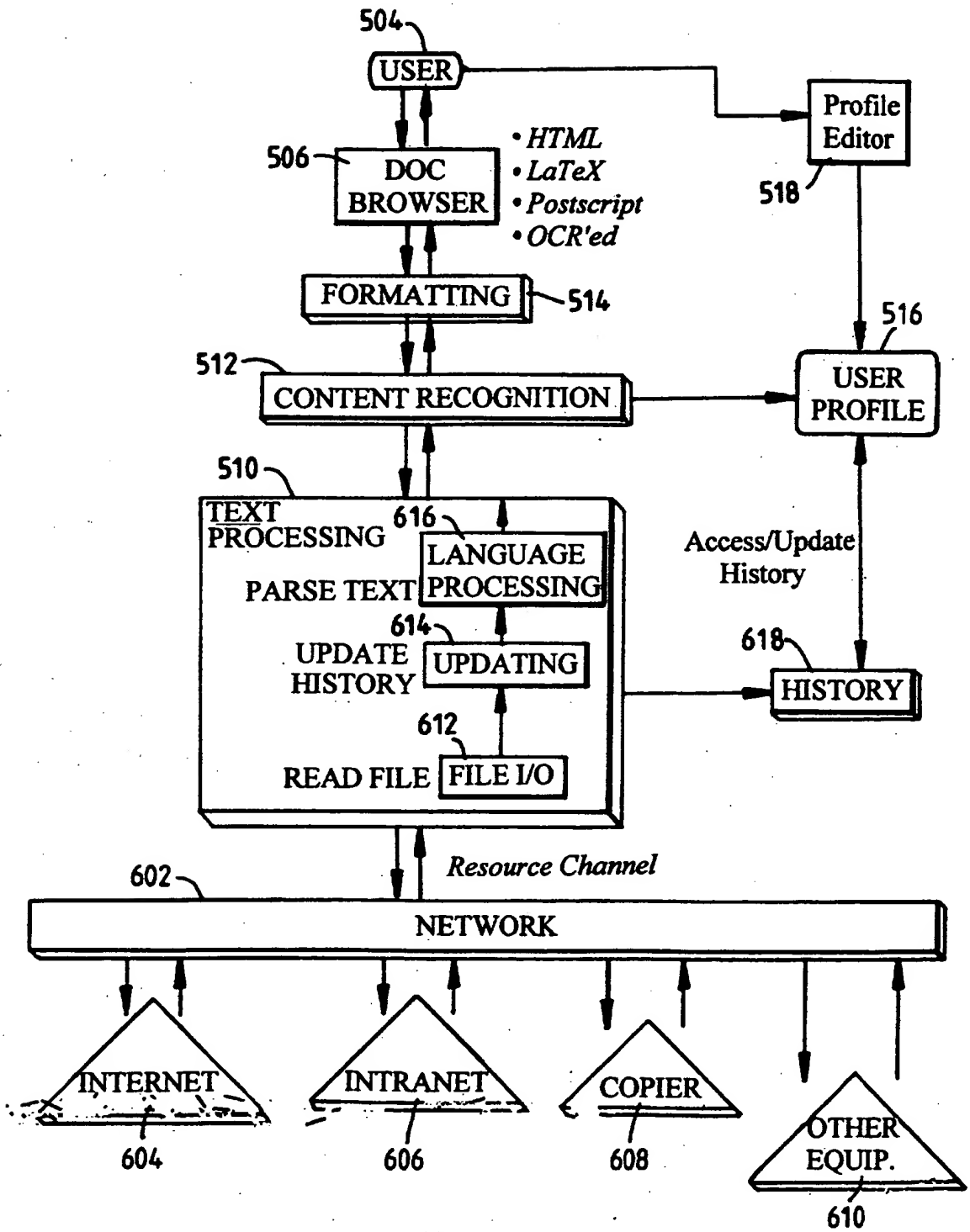

504
USER
Profile Editor
506
DOC BROWSER
• HTML
• LaTeX
• Postscript
• OCR'ed
518
FORMATTING
514
516
512
CONTENT RECOGNITION
USER PROFILE
510
TEXT PROCESSING
616
LANGUAGE PROCESSING
PARSE TEXT
Access/Update History
614
UPDATE HISTORY
UPDATING
618
HISTORY
612
READ FILE
FILE I/O
Resource Channel
602
NETWORK
INTERNET
INTRANET
COPIER
OTHER EQUIP.
604
606
608
610

Fig. 6A

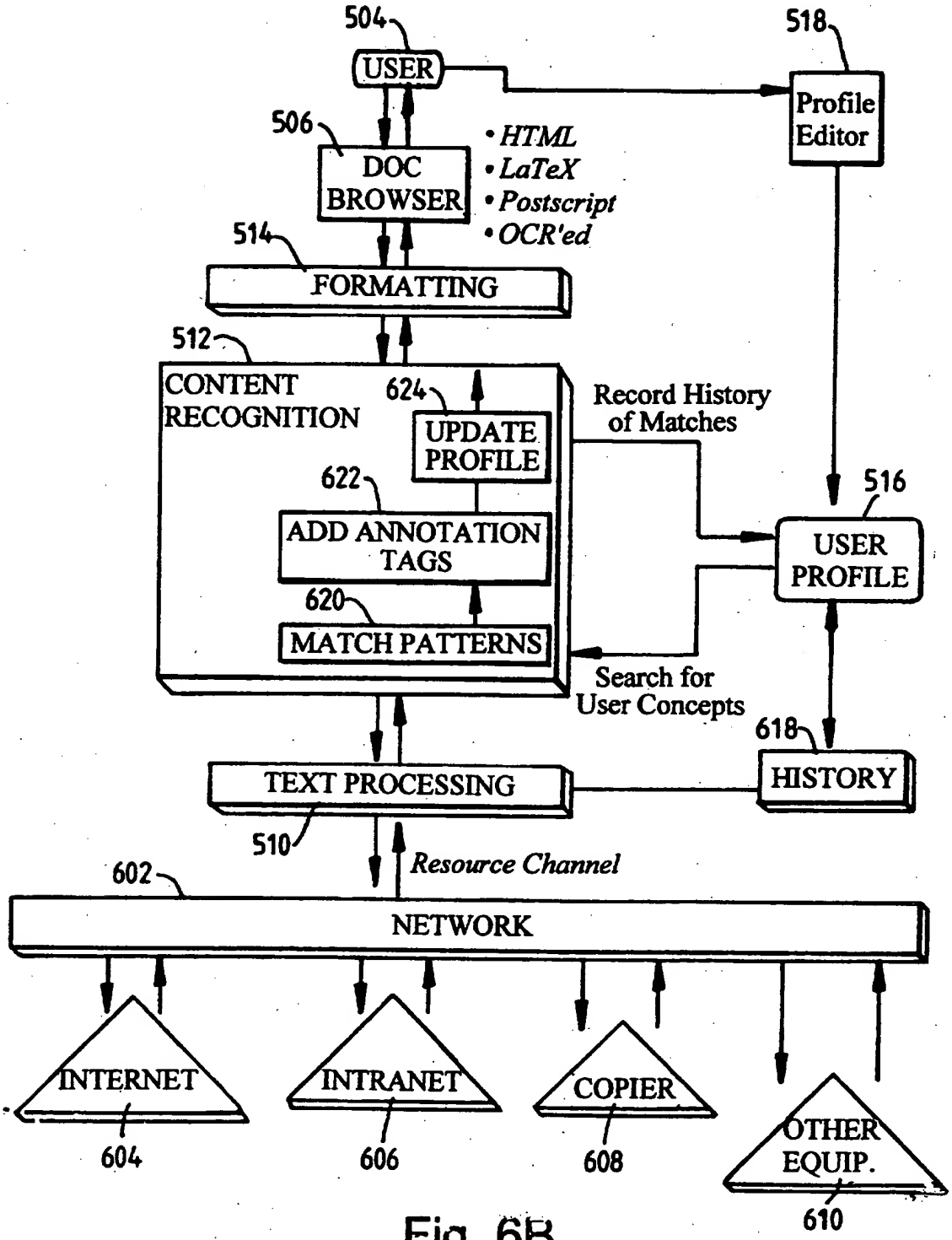

504
USER
518
Profile Editor
506
DOC BROWSER
• HTML
• LaTeX
• Postscript
• OCR'ed
514
FORMATTING
512
CONTENT RECOGNITION
624
UPDATE PROFILE
Record History of Matches
622
ADD ANNOTATION TAGS
516
USER PROFILE
620
MATCH PATTERNS
Search for User Concepts
618
HISTORY
TEXT PROCESSING
510
Resource Channel
602
NETWORK
INTERNET
604
INTRANET
606
COPIER
608
OTHER EQUIP.
610

Fig. 6B

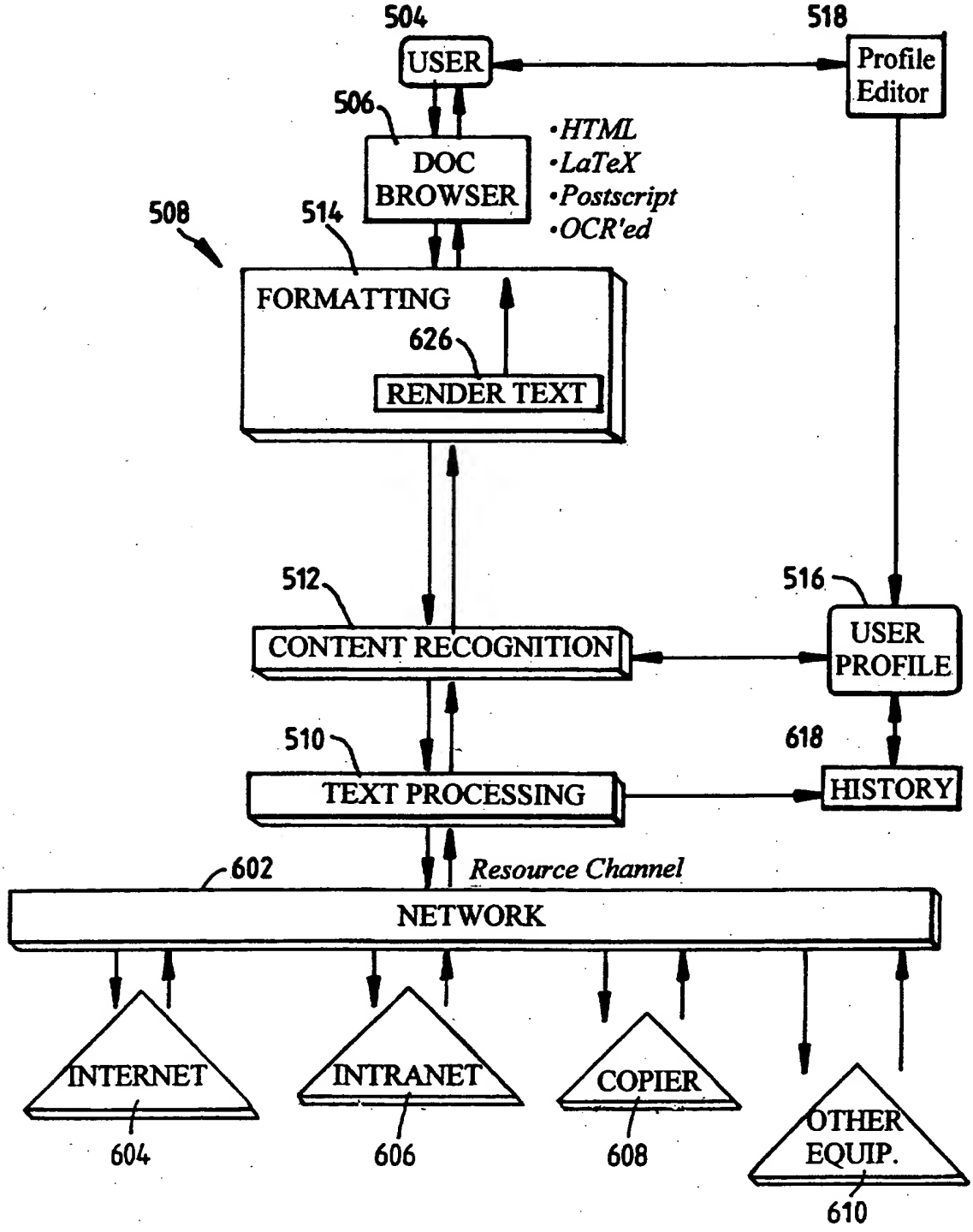

504
USER
518
Profile Editor
506
DOC BROWSER
•HTML
•LaTeX
•Postscript
•OCR'ed
508
514
FORMATTING
626
RENDER TEXT
512
CONTENT RECOGNITION
516
USER PROFILE
510
TEXT PROCESSING
618
HISTORY
602
Resource Channel
NETWORK
INTERNET
604
INTRANET
606
COPIER
608
OTHER EQUIP.
610

Fig. 6C

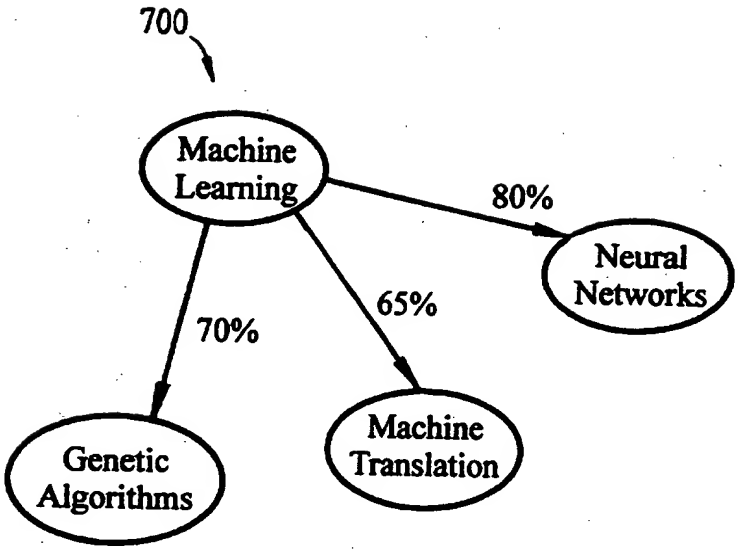

700
Machine Learning
80%
Neural Networks
70%
65%
Genetic Algorithms
Machine Translation

Fig. 7

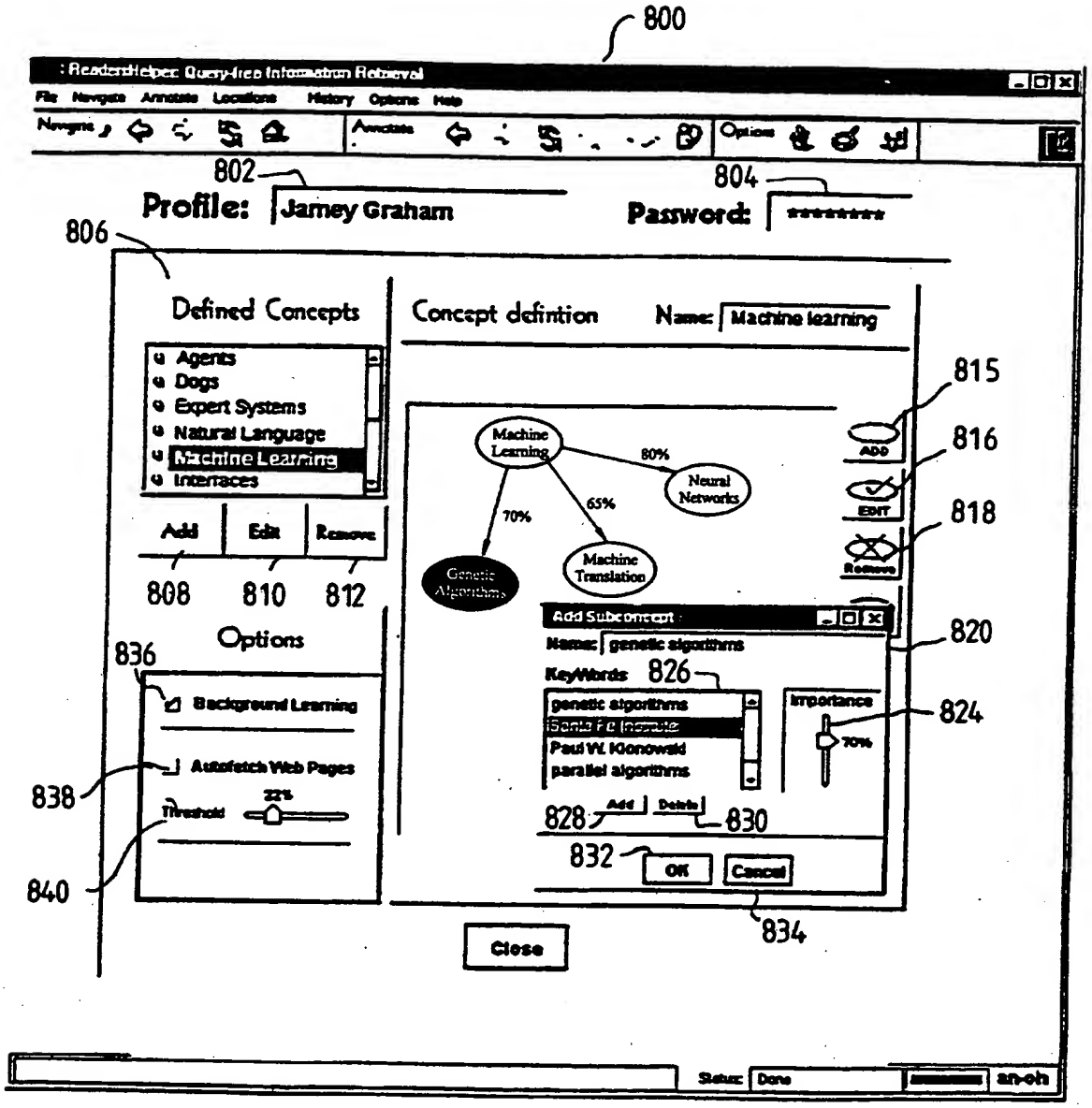

800
802
Profile:
Jamey Graham
804
Password:
806
Defined Concepts
Agents
Dogs
Expert Systems
Natural Language
Machine Learning
Interfaces
Add
Edit
Remove
808
810
812
Concept defintion
Name:
Machine learning
Machine Learning
80%
Neural Networks
70%
65%
Genetic Algorithms
Machine Translation
815
816
818
820
824
826
828
830
832
834
836
838
840
Options
Background Learning
Autofetch Web Pages
Threshold
22%
Add Subconcept
genetic algorithms
KeyWords
genetic algorithms
Paul W. Kronowald
parallel algorithms
Importance
70%
Add
Delete
OK
Cancel
Close
Status:
Done

Fig. 8

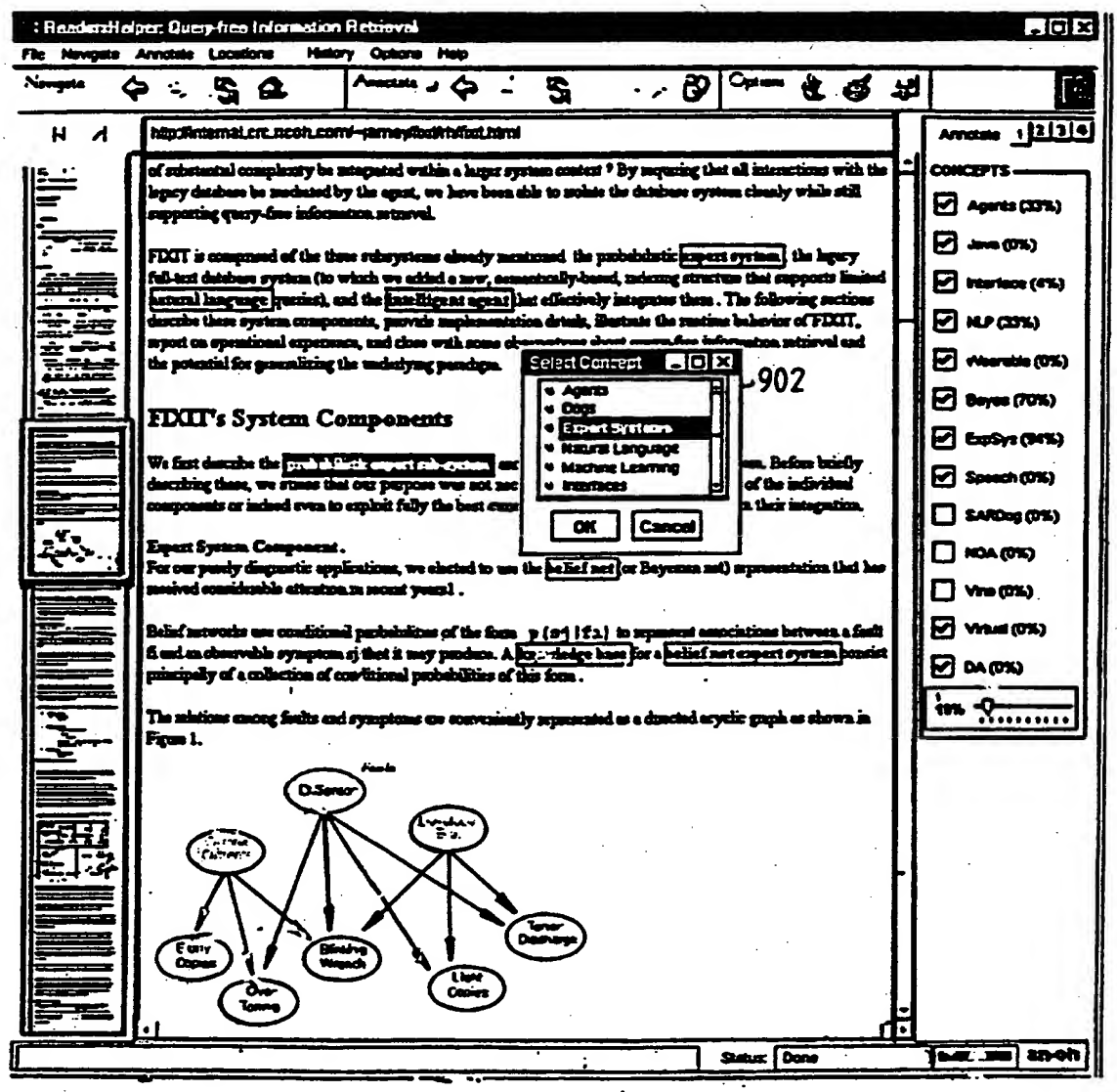

Reader Helper: Query-free Information Retrieval
File Navigate Annotate Locations History Options Help
Navigate
Annotate
Options
CONCEPTS
Agents (33%)
Java (0%)
Interface (0%)
NLP (33%)
Wearable (0%)
Bayes (70%)
ExpSys (94%)
Speech (0%)
SARDog (0%)
NOA (0%)
Vine (0%)
Virtual (0%)
DA (0%)
Select Concept
Agents
Oops
Expert Systems
Natural Language
Machine Learning
Interfaces
OK
Cancel
FIXIT's System Components
Expert System Component.
D-Sensor
Status: Done

Fig. 9A

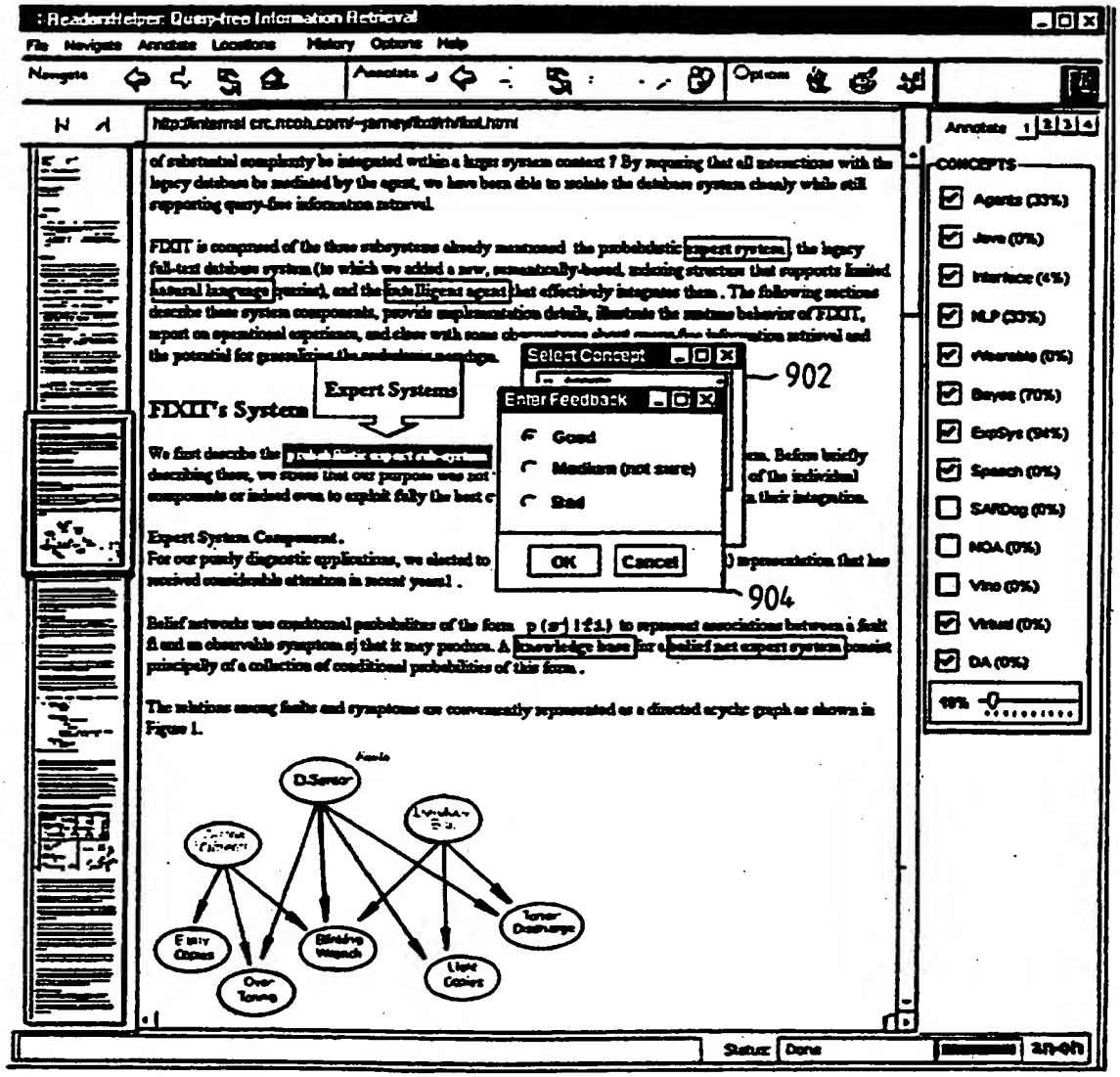
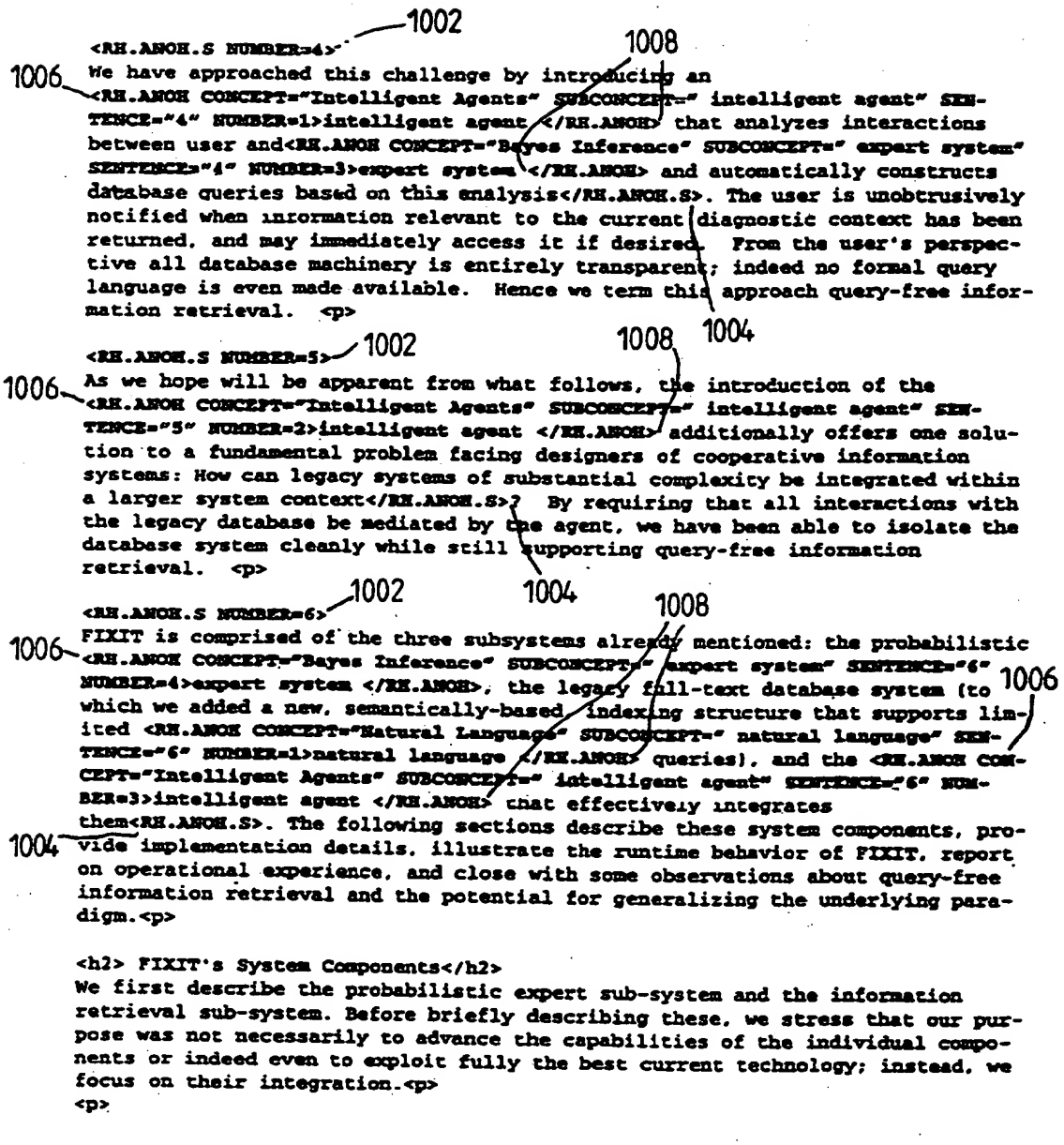

ReadersHelper: Query-free Information Retrieval
File Navigate Annotate Locations History Options Help
Navigate
Annotate
Options
Annotate 1 2 3 4
CONCEPTS
Agents (33%)
Java (0%)
Interface (4%)
NLP (33%)
Bayes (70%)
ExpSys (94%)
Speech (0%)
NOA (0%)
Vino (0%)
Virtual (0%)
DA (0%)
Status: Done
of substantial complexity be integrated within a larger system context ? By requiring that all interactions with the legacy database be mediated by the agent, we have been able to isolate the database system closely while still supporting query-free information retrieval.
FIXIT is composed of the three subsystems already mentioned: the probabilistic expert system, the legacy full-text database system (to which we added a new, semantically-based, indexing structure that supports limited natural language parsing), and the intelligent agent that effectively integrates them. The following sections describe these system components, provide implementation details, illustrate the runtime behavior of FIXIT, report on operational experience, and close with some observations about cooperative information retrieval and the potential for generalizing the architecture's paradigm.
Expert Systems
FIXIT's System
Select Concept
902
Enter Feedback
Good
Medium (not sure)
Bad
OK
Cancel
904
We first describe the
describing these, we stress that our purpose was not
components or indeed even to exploit fully the best
of the individual
in their integration.
Expert System Component.
For our purely diagnostic applications, we elected to
) representation that has
received considerable attention in recent years1 .
Belief networks use conditional probabilities of the form p(si | fi) to represent associations between a fault fi and an observable symptom sj that it may produce. A knowledge base for a belief net expert system consist principally of a collection of conditional probabilities of this form.
The relations among faults and symptoms are conveniently represented as a directed acyclic graph as shown in Figure 1.
D.Sensor
Engine Cranks
Battery Voltage
Lights Dashboard
Lights Cabin
Over Temp

Fig. 9B



<RH.ANOH.S NUMBER=4>
We have approached this challenge by introducing an
<RH.ANOH CONCEPT="Intelligent Agents" SUBCONCEPT=" intelligent agent" SENTENCE="4" NUMBER=1>intelligent agent </RH.ANOH> that analyzes interactions between user and<RH.ANOH CONCEPT="Bayes Inference" SUBCONCEPT=" expert system" SENTENCE="4" NUMBER=3>expert system </RH.ANOH> and automatically constructs database queries based on this analysis</RH.ANOH.S>. The user is unobtrusively notified when information relevant to the current diagnostic context has been returned, and may immediately access it if desired. From the user's perspective all database machinery is entirely transparent; indeed no formal query language is even made available. Hence we term this approach query-free information retrieval. <p>

<RH.ANOH.S NUMBER=5>
As we hope will be apparent from what follows, the introduction of the
<RH.ANOH CONCEPT="Intelligent Agents" SUBCONCEPT=" intelligent agent" SENTENCE="5" NUMBER=2>intelligent agent </RH.ANOH> additionally offers one solution to a fundamental problem facing designers of cooperative information systems: How can legacy systems of substantial complexity be integrated within a larger system context</RH.ANOH.S>? By requiring that all interactions with the legacy database be mediated by the agent, we have been able to isolate the database system cleanly while still supporting query-free information retrieval. <p>

<RH.ANOH.S NUMBER=6>
FIXIT is comprised of the three subsystems already mentioned: the probabilistic
<RH.ANOH CONCEPT="Bayes Inference" SUBCONCEPT=" expert system" SENTENCE="6" NUMBER=4>expert system </RH.ANOH>, the legacy full-text database system (to which we added a new, semantically-based indexing structure that supports limited <RH.ANOH CONCEPT="Natural Language" SUBCONCEPT=" natural language" SENTENCE="6" NUMBER=1>natural language </RH.ANOH> queries), and the <RH.ANOH CONCEPT="Intelligent Agents" SUBCONCEPT=" intelligent agent" SENTENCE="6" NUMBER=3>intelligent agent </RH.ANOH> that effectively integrates them<RH.ANOH.S>. The following sections describe these system components, provide implementation details, illustrate the runtime behavior of FIXIT, report on operational experience, and close with some observations about query-free information retrieval and the potential for generalizing the underlying paradigm.<p>

<h2> FIXIT's System Components</h2>
We first describe the probabilistic expert sub-system and the information retrieval sub-system. Before briefly describing these, we stress that our purpose was not necessarily to advance the capabilities of the individual components or indeed even to exploit fully the best current technology; instead, we focus on their integration.<p>
<p>

Fig. 10

AUTOMATIC ADAPTIVE DOCUMENT HELP SYSTEM

The present invention relates to display of electronic documents and more particularly to method and apparatus for augmenting electronic document display with features to enhance the experience of reading an electronic document on a display.

Increasingly, readers of documents are being called upon to assimilate vast quantities of information in a short period of time. To meet the demands placed upon them, readers find they must read documents "horizontally," rather than "vertically," i.e., they must scan, skim, and browse sections of interest in multiple documents rather than read and analyze a single document from beginning to end.

Documents are now more and more available in electronic form. Some documents are available electronically by virtue of their having been locally created using word processing software. Other electronic documents are accessible via the Internet. Yet others may become available in electronic form by virtue of being scanned in, copied, or faxed. See commonly assigned U.S. Application No. 08/754,721, entitled AUTOMATIC AND TRANSPARENT DOCUMENT ARCHIVING, the contents of which are herein incorporated by reference.

However, the mere availability of documents in electronic form does not assist the reader in confronting the challenges of assimilating information quickly. Indeed, many time-challenged readers still prefer paper documents because of their portability and the ease of flipping through pages.

Certain tools exist to take advantage of the electronic form documents to assist harried readers. Tools exist to search for documents both on the Internet and locally. However, once the document is identified and retrieved, further search capabilities are limited to keyword searching. Automatic summarization techniques have also been developed but have limitations in that they are not personalized. They summarize based on general features found in sentences.

What is needed is a document display system that helps the reader find as well as assimilate the information he or she wants more quickly. The document display system should be easily personalizable and flexible as well.

An automatic reading assistance application for documents in electronic form is provided by virtue of the present invention. In certain embodiments, an automatic annotator is provided which finds concepts of interest and keywords. The operation of the annotator is personalizable for a particular user. The annotator is also capable of improving its performance overtime by both automatic and manual feedback. The annotator is usable with any electronic document. Another available feature is a elongated thumbnail image of all or part of a multi-page document wherein a currently displayed section of the document is emphasized in the elongated thumbnail image. Movement of the emphasized area in the elongated thumbnail image is then synchronized with scrolling through the document.

In accordance with a first aspect of the present invention, a method for annotating an electronically stored document includes steps of: accepting user input indicating user-specific concepts of interest, analyzing the electronic document to identify locations of discussion of the user-specific concepts of interest, and displaying the electronic document with visual indications of the identified locations.

In accordance with a second aspect of the present invention, a method for displaying a multi-page document includes steps of: displaying a elongated thumbnail image of a multi-page document in a first viewing area of a display, displaying a section of the multi-page document in a second viewing area of the display in legible form, emphasizing an area of the elongated thumbnail image corresponding to the section displayed in the second viewing area, accepting user input controlling sliding of the emphasized area through the thumbnail image, and scrolling the displayed section through the second viewing area responsive to the scrolling so that the emphasized area continues to correspond to the displayed section.

A further understanding of the nature and advantages of the inventions herein may be realized by reference to the remaining portions of the specification and the attached drawings, in which:

Fig. 1 depicts a representative computer system suitable for implementing the present invention.

Figs. 2A-2D depict document browsing displays in accordance with one embodiment of the present invention.

Fig. 3 depicts a document summary view in accordance with one embodiment of the present invention.

Fig. 4 depicts a table of contents view in accordance with one embodiment of the present invention.

Fig. 5 depicts a top-level software architectural diagram for automatic annotation in accordance with one embodiment of the present invention.

Figs. 6A-6C depict a detailed software architectural diagram for automatic annotation in accordance with one embodiment of the present invention.

Fig. 7 depicts a representative Bayesian belief network useful in automatic annotation in accordance with one embodiment of the present invention.

Fig. 8 depicts a user interface for defining a user profile in accordance with one embodiment of the present invention.

Figs. 9A-9B depict an interface for providing user feedback in accordance with one embodiment of the present invention.

Fig. 10 depicts a portion of an HTML document processed in accordance with one embodiment of the present invention.

Computer System Usable for Implementing the Present Invention

Fig. 1 depicts a representative computer system suitable for implementing the present invention. Fig. 1 shows basic subsystems of a computer system 10 suitable for use with the present invention. In Fig. 1, computer system 10 includes a bus 12 which interconnects major subsystems such as a central processor 14, a system memory 16, an input/output controller 18, an external device such as a printer 20 via a parallel port 22, a display screen 24 via a display adapter 26, a serial port 28, a keyboard 30, a fixed disk drive 32 and a floppy disk drive 33 operative to receive a floppy disk 33A. Many other devices may be connected such as a scanner 34 via I/O controller 18, a mouse 36

connected to serial port 28 or a network interface 40. Many other devices or subsystems (not shown) may be connected in a similar manner. Also, it is not necessary for all of the devices shown in Fig. 1 to be present to practice the present invention, as discussed below. The devices and subsystems may be interconnected in different ways from that shown in Fig. 1. The operation of a computer system such as that shown in Fig. 1A is readily known in the art and is not discussed in detail in the present application. Source code to implement the present invention may be operably disposed in system memory 16 or stored on storage media such as a fixed disk 32 or a floppy disk 33A. Image information may be stored on fixed disk 32.

Annotated Document User Interface

The present invention provides a personalizable system for automatically annotating documents to locate concepts of interest to a particular user. Fig. 2A depicts one user interface 200 for viewing a document that has been annotated in accordance with the present invention. A first viewing area 202 shows a section of an electronic document. Using a scroll bar 204, or in other ways; the user may scroll the displayed section through the electronic document.

A series of concept check boxes 206 permit the user to select which concepts of interest are to be noted in the document. A sensitivity control 208 permits the user to select the degree of sensitivity to apply in identifying potential locations of relevant discussion. At low sensitivity, more locations will be denoted as being relevant, even though some may not be of any actual interest. At high sensitivity, most all denoted locations will in fact be relevant but some other relevant locations may be missed. After each concept name appearing by one of checkboxes 206 appears a percentage giving the relevance of the currently viewed document to the concept. These relevance levels offer a quick assessment of the relevance of the document to the selected concepts. Fig. 2A shows no annotations because a plain text view rather than an annotated view has been selected for first viewing area 202.

A thumbnail view 214 of the entire document is found in a second viewing area 215. Details of thumbnail view 214 will be discussed in greater detail below.

Miscellaneous navigation tools are found on a navigation toolbar 216.

Miscellaneous annotation tools are found on an annotation toolbar 218. The annotation tools on annotation toolbar 218 facilitate navigation through a collection of documents.

According to the present invention, annotations may be added to the text displayed in first viewing area 204. The annotations denote text relevant to user-selected concepts. As will be explained further below, an automatic annotation system according to the present invention adds these annotations to any document available in electronic form. The document need not include any special information to assist in locating discussion of concepts of interest.

Fig. 2B depicts the document view of Fig. 2A but with annotation added in first viewing area 202. Phrases 220 have been highlighted to indicate that they relate to concepts of interest to the user. The highlighting is preferably color. However, for ease of illustration in black-and-white format, rectangles indicate the highlighted areas of text. For further emphasis, the highlighted text is preferably printed in bold. A rectangular bar 222 indicates a paragraph that has been determined to have relevance above a predetermined threshold or to have more than a threshold number of key phrases. Rectangular bar 222 is merely representative of various forms of marginal annotation that might be used to indicate a relevant section of the text.

Fig. 2C depicts an alternative style of annotation. Now in first viewing area 202, entire sentences 224 including phrases relevant to concepts of interest are highlighted. The phrases themselves are printed in bold text. It has been found that highlighting the entire sentence rather than just a relevant phrase provides the user with far more information at a glance.

Fig. 2D depicts how further information about key phrases may be displayed. The user may select any highlighted key phrase with the mouse. Upon selection of the key phrase, a balloon 226 appears. The balloon includes further information relevant to the key phrase. For example, the balloon may include the name of the concept to which the keyword is relevant. The balloon may also include bibliographic information if the key phrase includes a citation.

Fig. 3 depicts a document summary view in accordance with one embodiment of the present invention. The user may optionally select a summary view 300 of the document. Summary view lists the concepts of interest 302 that are found in the documents as headings of an outline. For each concept, keywords or key phrases 304 are

listed which are indicative of the concept of interest. A number in parenthesis by each keyword indicates the number of times the keyword or key phrase appears. Each concept also has an associated score 306 indicative of the relevance of the whole document to the concept.

Fig. 4 depicts a table of contents view in accordance with one embodiment of the present invention. An alternative to summary view 300 is a table of contents view 400. Table of contents view 400 lists major headings 402 and subheadings 403 of the electronic document. By selecting one of hierarchical display icons 404, the user may list the concepts 406 found under one of the document headings 402 or subheadings 403 with an indication of relevance for each concept and the number of keywords found. There is also a relevance meter 408 for each document heading 402 that indicates the overall relevance of the text under that heading for all of the currently selected concepts. In a preferred embodiment where the document is an HTML document, to create table-of-contents view 400, the headings of the document are identified by an analysis of the HTML heading tags.

Automatic Annotation Software

Fig. 5 depicts a top-level software architectural diagram for automatic annotation in accordance with one embodiment of the present invention. A document 502 exists in electronic form. It may have been scanned in originally. It may be, e.g., in HTML, Postscript, LaTeX, other word processing or e-mail formats, etc. The description that follows assumes an HTML format. A user 504 accesses document 502 through a document browser 506 and an annotation agent 508. Document browser 506 is preferably a hypertext browsing program such as Netscape Navigator or Microsoft Explorer but also may be, e.g., a conventional word processing program.

Annotation agent 508 adds the annotations to document 502 to prepare it for viewing by document browser 506. Processing by annotation agent 508 may be understood to be in three stages, a text processing stage 510, a content recognition stage 512, and a formatting stage 514. The input to text processing stage 510 is raw text. The output from text processing stage 510 and input to content recognition stage 512 is a parsed text stream, a text stream with formatting information such as special tags around particular words or phrases removed. The output from content recognition stage 512 and

input to formatting stage 514 is an annotated text stream. The output of formatting stage 514 is a formatted text file viewable with document browser 506.

The processing of annotation agent 508 is preferably a run-time process. The annotations are not preferably pre-inserted into the text but are rather generated when user 504 requests document 502 for browsing. Thus, this is preferably a dynamic process. Annotation agent 508 may also, however, operate in the background as a batch process.

The annotation added by annotation agent 508 depends on concepts of interest selected by user 504. User 504 also inputs information used by annotation agent 508 to identify locations of discussion of concepts of interest in document 502. In a preferred embodiment, this information defines the structure of a Bayesian belief network. The concepts of interest and other user-specific information are maintained in a user profile file 516. User 504 employs a profile editor 518 to modify the contents of user profile file 516.

Fig. 6A depicts the automatic annotation software architecture of Fig. 5 with text processing stage 510 shown in greater detail. Fig. 6A shows that the source of document 502 may be accessed via a network 602. Possible sources include e.g., the Internet 604, an intranet 606, a digital copier 608 that captures document images, or other office equipment 610 such as a fax machine, scanner, printer, etc. Another alternative source is the user's own hard drive 32.

Text processing stage 510 includes a file I/O stage 612, an updating stage 614, and a language processing stage 616. File I/O stage reads the document file from network 602. Updating stage 614 maintains a history of recently visited documents in a history file 618. Language processing stage 616 parses the text of document 502 to generate the parsed text output of text processing stage 510.

Fig. 6B depicts the automatic annotation software architecture of Fig. 5 with content recognition stage 512 shown in greater detail. A pattern identification stage 620 looks for particular patterns in the parsed text output of text processing stage 510. The particular patterns searched for are determined by the contents of user profile file 516. Once the patterns are found, annotation tags are added to the parsed text by an annotation tag addition stage 622 to indicate the pattern locations. In a preferred HTML embodiment, these annotation tags are compatible with the HTML format. However, the tagging process may be adapted to LaTeX, Postscript, etc. A profile updating stage 624 monitors

the output of annotation tag addition stage 622 and analyzes text surrounding the locations of concepts of interest. As will be further discussed with reference to Fig. 7 changes the contents of user profile file 516 based on the analysis of this surrounding text. The effect is to automatically refine the patterns searched for by pattern identification stage 620 to improve annotation performance.

Fig. 6C depicts the automatic annotation software architecture of Fig. 5 with formatting stage 514 shown in greater detail. Formatting stage 514 includes a text rendering stage 626 that formats the annotated text provided by content recognition stage 512 to facilitate viewing by document browser 506. An HTML document as modified by formatting stage 514 is discussed in greater detail with reference to Fig. 10.

Pattern identification stage 620 looks for keywords and key phrases of interest and locates relevant discussion of concepts based on the located keywords. The identification of keywords and the application of the keywords to locating relevant discussion is preferably accomplished by reference to a belief system. The belief system is preferably a Bayesian belief network.

Fig. 7 depicts a portion of a representative Bayesian belief network 700 implementing a belief system as used by pattern identification stage 622. A first oval 702 represents a particular user-specified concept of interest. Other ovals 704 represent subconcepts related to the concept identified by oval 702. Each line between one of subconcept ovals 704 and concept oval 702 indicates that discussion of the subconcept implies discussion of the concept. Each connection between one of subconcept ovals 704 and concept oval 702 has an associated probability value indicated in percent. These values in turn indicate the probability that the concept is discussed given the presence of evidence indicating the presence of the subconcept. Discussion of the subconcept is in turn indicated by one or more keywords or key phrases (not shown in Fig. 7).

The structure of Bayesian belief network 700 is only one possible structure applicable to the present invention. For example, one could employ a Bayesian belief network with more than two levels of hierarchy so that the presence of subconcepts is suggested by the presence of "subsubconcepts" and so on. In the preferred embodiment, presence of a keyword or key phrase always indicates presence of discussion of the subconcept but it is also possible to configure the belief network so that presence of a keyword or key phrase suggests discussion of the subconcept with a specified probability.

The primary source for the structure of Bayesian belief network 700 including the selection of concepts, keywords and key phrases, interconnections, and probabilities is user profile file 516. In a preferred embodiment, user profile file 516 is selectable for both editing and use from among profiles for many users.

The structure of belief system 700 is however also modifiable during use of the annotation system. The modifications may occur automatically in the background or may involve explicit user feedback input. The locations of concepts of interest determined by pattern identification stage 620 are monitored by profile updating stage 624. Profile updating stage 624 notes the proximity of other keywords and key phrases within each analyzed document to the locations of concepts of interest. If particular keywords and key phrases are always near a concept of interest, the structure and contents of belief system 700 are updated in the background without user input by profile updating stage 624. This could mean changing probability values, introducing a new connection between a subconcept and concept, or introducing a new keyword or key phrase.

User 504 may select a word or phrase in document 502 as being relevant to a particular concept even though the word or phrase has not yet defined to be a keyword or key phrase. Belief system 700 is then updated to include the new keyword or key phrase

User 504 may also give feedback for an existing key word or key phrase, indicating the perceived relevance of the keyword or key phrase to the concept of interest. If the selected keyword or key phrase is indicated to be of high relevance to the concept of interest, the probability values connecting the subconcept indicated by the selected keywords or key phrases to the concept of interest increases. If, on the other hand, user 504 indicates the selected keywords or key phrases to be of little interest, the probability values connecting these keywords or key phrases to the concept decrease.

User Profile and Feedback Interfaces

Fig. 8 depicts a user interface for defining a user profile in accordance with one embodiment of the present invention. User interface screen 800 is provided by profile editor 518. A profile name box 802 permits the user to enter the name of the person or group to whom the profile to be edited is assigned. This permits the annotation system according to the present invention to be personalized to particular users or groups. A

password box 804 provides security by requiring entry of a correct password prior to profile editing operations.

A defined concepts list 806 lists all of the concepts which have already been added to the user profile. By selecting a concept add button 808, the user may add a new concept. By selecting a concept edit button 810, the user may modify the belief network as it pertains to the listed concept that is currently selected. By selecting a remove button 812, the user may delete a concept.

If a concept has been selected for editing, its name appears in a concept name box 813. The portion of the belief network pertaining to the selected concept is shown in a belief network display window 814. Belief network display window 814 shows the selected concept, the subconcepts which have been defined as relating to the selected concept and the percentage values associated with each relationship. The user may add a subconcept by selecting a subconcept add button 815. The user may edit a subconcept by selecting the subconcept in belief network display window 814 and then selecting a subconcept edit button 816. A subconcept remove button 818 permits the user to delete a subconcept from the belief network.

Selecting subconcept add button 815 causes a subconcept add window 820 to appear. Subconcept add window 820 includes a subconcept name box 822 for entering the name of a new subconcept. A slider control 824 permits the user to select the percentage value that defines the probability of the selected concept appearing given that the newly selected subconcept appears. A keyword list 826 lists the keywords and key phrases which indicate discussion of the subconcept. The user adds to the list by selecting a keyword add button 828 which causes display of a dialog box (not shown) for entering the new keyword or key phrase. The user deletes a keyword or key phrase by selecting it and then selecting a keyword delete button 830. Once the user has finished defining the new subconcept, he or she confirms the definition by selecting an OK button 832. Selection of a cancel button 834 dismisses subconcept add window 820 without affecting the belief network contents or structure. Selection of subconcept edit button 816 causes display of a window similar to subconcept add window 820 permitting redefinition of the selected subconcept.

By selecting whether a background learning checkbox 836 has been selected, the user may enable or disable the operation of profile updating stage 624. A

web autofetch check box 838 permits the user to select whether or not to enable an automatic web search process. When this web search process is enabled, whenever a particular keyword or key phrase is found frequently near where a defined concept is determined to be discussed, a web search tool such as AltaVista™ is employed to look on the World Wide Web for documents containing the keyword or key phrase. A threshold slider control 840 is provided to enable the user to set a threshold relevance level for this autofetching process.

Figs. 9A-9B depict a user interface for providing feedback in accordance with one embodiment of the present invention. User 502 may select any text and call up a first feedback window 902. The text may or may not have been previously identified by the annotation system as relevant. In first feedback window 902 shown in Fig. 9A, user 504 may indicate the concept to which the selected text is relevant. First feedback window 902 may not be necessary when adjusting the relevance level for a keyword or key phrase that is already a part of belief network 700. After the user selects a concept in first feedback window 902, a second feedback window 904 is displayed for selecting the degree of relevance. Second feedback window 904 in Fig. 9B provides three choices for level of relevance: good, medium (not sure), and bad. Alternatively, a slider control could be used to set the level of relevance. If the selected text is not already a keyword or key phrase in belief network 700, a new subconcept is added along with the associated new keyword or key phrase. If the selected text is already a keyword or key phrase, above, probability values within belief system 622 are modified appropriately in response to this user feedback.

Fig. 10 depicts a portion of an HTML document 1000 processed in accordance with one embodiment of the present invention. A sentence including relevant text is preceded by an a <RH.ANOH.S ...> tag 1002 and followed by an </RH.ANOH.S> tag 1004. The use of these tags facilitates the annotation mode where complete sentences are highlighted. The <RH.ANOH.S ...> tag 1002 includes a number indicating which relevant sentence is tagged in order of appearance in the document. Relevant text within a so-tagged relevant sentence is preceded by an <RH.ANOH ... > tag 1006 and followed by an </RH.ANOH> tag 1008. The <RH.ANOH ... > 1006 tag include the names of the concept and subconcept to which the annotated text is relevant, an identifier indicating which relevant sentence the text is in and a number which

identifies which annotation this is in sequence for a particular concept. An HTML browser that has not been modified to interpret the special annotation tags provided by the present invention will ignore them and display the document without annotations.

Thumbnail Image Display

Referring again to Figs. 2A-2D, an elongated thumbnail image 214 of many pages, or all of document 502 is presented in second viewing area 215. Document 502 will typically be a multi-page document with a section being displayed in first viewing area 202. Elongated thumbnail image 214 provides a convenient view of the basic document structure. The annotations incorporated into the document are visible within elongated thumbnail image 214. Within elongated thumbnail image 214, an emphasized area 214A shows a reduced view of the document section currently displayed in first viewing area 215 with the reduction ratio preferably being user-configurable. Thus, if the first viewing area 202 changes in size because of a change of window size, emphasized area 214A will also change in size accordingly. The greater the viewing area allocated to elongated thumbnail image 214 and emphasized area 214A, the more detail is visible. With very small allocated viewing areas, only sections of the document may be distinguishable. As the allocated area increases, individual lines and eventually individual words become distinguishable. In Figs. 2A-2D the user-configured ratio is approximately 5:1. Emphasized viewing area 214 may be understood to be a lens or a viewing window over the part of elongated thumbnail image 214A corresponded to the document section displayed in first viewing area 215. User 504 may scroll through document 502 by sliding emphasized area 214A up and down. As emphasized area 214A shifts, the section of document 502 displayed in first viewing area 202 will also shift. User 504 may also scroll conventionally using scroll bar 204 or arrow keys and emphasized area 214A will slide up or down as appropriate in response.

In Figs. 2A-2C elongated thumbnail image 214 displays each page of document 502 as being displayed at the same reduced scale. The present invention also contemplates other modes of scaling elongated thumbnail image 214. For example, one may display emphasized area 214A at a scale similar to that shown in Figs. 2A-2C and use a variable scale for the rest of elongated thumbnail image 214. Text from far away

emphasized area 214A would be displayed at a highly reduced scale and the degree of magnification would increase with nearness to emphasized area 214A.

Because, the annotations appear in enlongated thumbnail image 214, it is very easy to find relevant text anywhere in document 502. Furthermore, elongated thumbnail image 214 provides a highly useful way of keeping track of one's position within a lengthy document.

Software Implementation

In a preferred embodiment, software to implement the present invention is written in the Java language. Preferably, the software forms a part of a stand-alone browser program written in the Java language. Alternatively, the code may be in the form of a so-called "plug-in" operating with a Java-equipped web browser used to browse HTML documents including the special annotation tags explained above.

In the foregoing specification, the invention has been described with reference to specific exemplary embodiments thereof. For example, any probabilistic inference method may be substituted for a Bayesian belief network. It will, however, be evident that various modifications and changes may be made thereunto without departing from the broader spirit and scope of the invention as set forth in the appended claims and their full scope of equivalents.

CLAIMS:

1. An automatic adaptive document help system for annotating an electronically stored document, the system comprising:

means for storing a user-specified concept of interest;

means for locating discussion of said concept of interest within the electronically stored document; and

means for displaying said electronic document with visual indications of said identified locations.

2. A computer-implemented method for annotating an electronically stored document comprising the steps of:

accepting user input indicating a user-specified concept of interest;

analyzing said electronic document to identify locations of discussion of said user-specified concept of interest; and

displaying said electronic document with visual indications of said identified locations.

3. The method of claim 2 wherein said analyzing step comprises exploiting a probabilistic inference method to identify said locations.

4. The method of claim 3 wherein said probabilistic inference method comprises a Bayesian belief network.

5. The method of claim 4 further comprising the step of:

accepting user input defining a structure of said Bayesian belief network.

6. The method of claim 4 or claim 5 further comprising the step of:

modifying said Bayesian belief network in accordance with content of previously visited electronic documents.

7. The method of any one of the claims 2 to 6 wherein said displaying step comprises the substep of:

highlighting sections of said document surrounding said locations.

8 The method of any one of the claims 2 to 6 wherein said displaying step comprises the substep of:

displaying a balloon pointing to a user-selected one of said locations, said balloon identifying said user-specified concept to which text in said user-selected one of said locations is relevant.

9. The method of any one of the claims 2 to 6 wherein said displaying step comprises the substep of:

displaying marginal notation identifying said locations.

10. The method of claim 4 or claim 5 further comprising the steps of:

accepting user input indicating a degree of relation between said locations and said concept of interest; and

modifying said Bayesian belief network responsive to said degree of relation.

11. The method of any one of the claims 2 to 10 further comprising the step of displaying a level of relevance of said document to said concept of interest.

12. A computer-implemented method for displaying a multipage document comprising the steps of:

displaying an elongated thumbnail image of a multi-page document in a first viewing area of a display;

displaying a section of said multi-page document in a second viewing area of said display in legible form;

emphasizing an area of said thumbnail image corresponding to said section displayed in said second viewing area;

accepting user input controlling sliding said emphasized area through said multi-page document; and

scrolling said displayed section in said second viewing area responsive to said sliding so that said emphasized area continues to correspond to said displayed section.

13. The method of claim 12 further comprising the steps of:

accepting user input indicating user-specific concepts of interest;

analyzing said multi-page document to identify locations of discussion of said user-specific concepts of interest;

marking said locations in both said thumbnail image and in said displayed section in said second viewing area.

14. A computer program product for annotating an electronically stored document comprising:

code for accepting user input indicating a user-specified concept of interest;

code for analyzing said electronic document to identify locations of discussion of said user-specified concepts of interest;

code for displaying said electronic document with visual indications of said identified locations; and

a computer-readable storage medium for storing the codes.

15. The product of claim 14 wherein said analyzing code comprises code for exploiting a probabilistic inference method to identify said locations.

16. The product of claim 15 wherein said probabilistic inference method comprises a Bayesian belief network.

17. The product of claim 16 further comprising code for:

accepting user input defining a structure of said Bayesian belief network.

18. The product of claim 16 or claim 17 further comprising code for modifying said Bayesian belief network in accordance with content of said electronic document.

19. The product of claim 18 wherein said modifying code comprises code for updating said Bayesian belief network in accordance with proximity of keywords to said identified locations.

20. The product of any one of the claims 14 to 19 wherein said displaying code comprises code for highlighting said locations.

21. The product of any one of the claims 14 to 19 wherein said displaying code comprises code for highlighting sections of said document surrounding said locations.

22. The product of any one of the claims 14 to 19 wherein said displaying code comprises code for displaying balloons pointing to said locations.

23. The product of any one of the claims 14 to 19 wherein said displaying code comprises code for displaying marginal notations identifying said locations.

24. The product of claim 16 or claim 17 further comprising:

code for accepting user input indicating a degree of relation between said locations and said concepts of interest; and

code for modifying said Bayesian belief network responsive to said degree of relation.

25. The product of any one of the claims 14 to 24 further comprising code for displaying a level of relevance of said document to said concept of interest.

26. A computer program product for displaying a multipage document comprising:

code for displaying an elongated thumbnail image of a multi-page document in a first viewing area of a display;

code for displaying a section of said multi-page document in a second viewing area of said display in legible form;

code for emphasizing an area of said thumbnail image corresponding to said section displayed in said second viewing area;

code for accepting user input controlling sliding of said emphasized area through said thumbnail image;

code for scrolling said displayed section so that said displayed section continues to correspond to said emphasized area; and

a computer-readable storage medium for storing the codes.

27. The computer program product of claim 26 further comprising:

code for accepting user input indicating user-specific concepts of interest;

code for analyzing said multi-page document to identify locations of discussion of said user-specific concepts of interest; and

code for marking said locations in both said thumbnail image and in said displayed section in said second viewing area.

28. A computer system comprising:

a processor; and

a computer-readable storage medium storing code to be executed by said processor, said code comprising:

code for accepting user input indicating user-specific concepts of interest;

code for analyzing an electronic document to identify locations of discussion of said user-specific concepts of interest; and

code for displaying said electronic document with visual indications of said identified locations.

29. A computer program product for displaying a multipage document comprising:

code for displaying an elongated thumbnail image of a multi-page document in a first viewing area of a display;

code for displaying a section of said multi-page document in a second viewing area of said display in legible form;

code for emphasizing an area of said thumbnail image corresponding to said section displayed in said second viewing area;

code for accepting user input indicating user-specific concepts of interest;

code for analyzing said multi-page document to identify locations of discussion of said user-specific concepts of interest; and

code for marking said locations in both said thumbnail image and in said displayed section in said second viewing area.

30. A computer system comprising:

a processor; and

a computer-readable storage medium storing code to be executed by said processor, said code comprising:

means for accepting user input indicating user-specific concepts of interest;

means for analyzing an electronic document to identify locations of discussion of said user-specific concepts of interest; and

means for displaying said electronic document with visual indications of said identified locations.

31. A computer system for displaying a multipage document comprising:

means for displaying an elongated thumbnail image of a multi-page document in a first viewing area of a display;

means for displaying a section of said multi-page document in a second viewing area of said display in legible form;

means for emphasizing an area of said thumbnail image corresponding to said section displayed in said second viewing area;

means for accepting user input indicating user-specific concepts of interest;

means for analyzing said multi-page document to identify locations of discussion of said user-specific concepts of interest; and

means for marking said locations in both said thumbnail image and in said displayed section in said second viewing area.